

REMARKS/ARGUMENTS

Pending claim 1 recites, in part, that a stack pointer for an operand tag stack be stored in a tag stack register in which the stack pointer is implicitly encoded. The claim also recites that the stack pointer is updated by movement of the stack pointer within the tag stack register. Furthermore, the claim recites that a bit position of this pointer is to indicate a depth of the operand tag stack and corresponds to a number of operand tags stored in the tag stack register. Still further, it is recited that a first significant bit of the tag stack register having a predetermined value serves as the stack pointer.

The cited art alone or in combination fails to teach or suggest at least this subject matter of claim 1. As to the primary reference, Ramesh, there is no teaching or suggestion of updating of a stack pointer by movement of the stack pointer to recognize addition of an operand tag. As contended support, the Office Action refers to FIG. 5B and columns 5 - 6 of Ramesh. As an initial matter, Ramesh fails to teach or suggest to the recited tag stack register that stores a stack of operand tags for operands stored in an operand stack of a memory. Instead, Ramesh is directed to a tag register that simply identifies a set of data registers. Ramesh, col. 3, lns. 9-14.

More so, while Ramesh teaches that a top of stack pointer can be incremented or decremented, there is absolutely no teaching or suggestion in the reference that update to the stack pointer is by movement of the pointer. Instead, Ramesh teaches the conventional increment/decrement of the pointer. The secondary references also fail to address this missing subject matter of Ramesh.

Here, the Office Action refers to the secondary reference Tran, which teaches the presence of a register map to point to one of multiple physical registers in a register stack. As with Ramesh, this map has nothing to do with the recited tag stack register, which stores a stack of operand tags for operands stored in an operand stack of a memory. Instead, Tran teaches a simple map for processor registers. Furthermore, Tran nowhere teaches the recited implicit encoding of a stack pointer that moves. Instead, all that the reference teaches is that a fixed pointer location in the register map can point to the top of these physical registers. Tran, col. 15, lns. 57-64.

The art further fails to teach or suggest the subject matter of claim 1 that a bit position of the stack pointer indicates a depth of the operand tag stack. Here, the Office Action again refers to Ramesh, stating that it would be obvious for this subject matter, given that the stack pointer is

incremented/decremented. This is simply not so. That is, regardless of the incrementing or decrementing of the stack pointer in Ramesh, its position does not change.

Claim 1 further describes that this movable bit position for the stack pointer corresponds to a number of operand tags stored in the tag stack register, and that a first significant bit of the tag stack register having a predetermined value serves as the stack pointer. None of the art anywhere teaches or suggests this subject matter of claim 1. As discussed above, Ramesh simply teaches that a stack pointer (which has nothing to do with the recited tag stack register) refers to a top of a register stack. The secondary reference Tran is also silent in this regard. Still further, the additional secondary reference Adl-Tabatabai only teaches the presence of a bit vector that can indicate a reference type of a corresponding variable. Nonetheless, nothing here anywhere teaches or suggests that a given significant bit of this bit vector have a particular value to serve as a stack pointer. Instead, each bit of the bit vector only indicates whether a variable is a reference or non-reference type. For all of these reasons, claim 1 and the claims depending therefrom are patentable over the cited art. For at least similar reasons, independent claims 6, 12, and 17 and their dependent claims are similarly patentable.

The application is believed to be in condition for allowance and the Examiner's prompt action in accordance therewith is respectfully requested. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 20-1504 (ITL2176US).

Respectfully submitted,

Date: January 19, 2010

/Mark J. Rozman/
Mark J. Rozman
Registration No. 42,117
TROP, PRUNER & HU, P.C.
1616 S. Voss Road, Suite 750
Houston, Texas 77057-2631
(512) 418-9944 [Phone]
(713) 468-8883 [Fax]
Customer No.: 21906